

**SOFTWARE ENGINE FOR COMBINING
VIDEO OR AUDIO CONTENT WITH PROGRAMMATIC CONTENT**

5

This patent document is a Continuation-in-part of U.S. Patent Application Serial Number (not yet assigned), filed on August 24, 2000 (Our Docket No. 68532, for SOFTWARE ENGINE FOR COMBINING VIDEO OR AUDIO CONTENT WITH PROGRAMMATIC CONTENT, which claims the benefit of U.S. Provisional Patent Application Serial No. 60/220,397, filed on July 24, 2000 for DVD CONTENT GENERATION USING AN XML INTERFACE, all of which are incorporated herein by reference.

15

BACKGROUND OF THE INVENTION

The present invention relates the authoring of video and/or audio content (referred to generically herein as video/audio content), and programmatic content for storage on or transmission through a medium, and more particularly to a software engine for authoring video and/or audio content, and programmatic content for storage on or transmission through a medium, such as an optical storage medium, or a computer network (such as through downloading of a multimedia file, or streaming of video or audio). Even more particularly, the present invention relates to a software engine for receiving a definition for video or audio content and for combining the definition with programmatic content developed in a development environment, and for outputting a ROM image or transmission image as a function of the definition and the programmatic content.

Software authoring tools in the computer software industry provide various mechanisms to assist the

software author in preparing software deliverables
suitable for use by an end user. The purpose of these
software-authoring tools is to provide a mechanism by
which otherwise tedious and repetitive tasks can be
5 streamlined. This may involve, for example, the
conversion of software source code of one type into
software source code of another type or the provision of
subroutine libraries that can be invoked by or included by
a software author in software deliverables in order to
10 effect a business method sought to be achieved through a
particular software implementation.

Recently, a great deal of attention has been
focused on the digital video disk (DVD) industry, with
deployment of DVD technology exceeding that of all prior
15 or similar technologies, e.g., audio cassette tapes, video
tapes, laser disks, compact disks and the like to a
significant degree.

The use of DVD technology on computers, such as
personal computers, has provided a combination that allows
20 a degree of interactivity to be achieved between DVD
content and an end user. Problematically, mechanisms for
controlling DVD hardware using computer software have been
quite primitive and limited in nature, and significant
interactivity has been restrained.

25 With the Internet now a further mechanism for
interactivity, the computer software industry's lack of
DVD interactivity mechanisms has only further been
highlighted.

The present invention advantageously addresses
30 the above and other needs.

SUMMARY OF THE INVENTION

The present invention advantageously addresses the needs above as well as other needs by providing a system and method for authoring video or audio content, and programmatic content for storage on or transmission through a medium.

In one embodiment, the present invention can be characterized as a method for combining video/audio content with programmatic content comprising the following steps: generating authoring output comprising a definition for a variable, and further comprising a representation of the video/audio content; selecting a source file, the source file comprising the variable; searching the source file for the variable, and replacing the variable with the definition for the variable; generating programmatic content in response to the searching; and generating an image as a function of the programmatic content and the representation of the audio/video content.

In a further embodiment, the present invention can be characterized as a system for combining video/audio content with programmatic content. The system searches a source file for a variable, and replaces the variable with a definition for the variable; generates programmatic content in response to the searching; and generates an image as a function of the programmatic content and a representation of the audio/video content.

In yet a further embodiment, the invention can be characterized as a system for combining video/audio content with programmatic content. The system employs a parser adapted to search a source file for a variable,

replace the variable with a definition for the variable,
and generate programmatic content in response to the
searching; and an image engine adapted to generate an
image as a function of the programmatic content and a
5 representation of the audio/video content.

In accordance with another embodiment, the
present invention can be characterized as a software
system employing a searching module for searching a source
file for a variable, and for replacing the variable with a
10 definition for the variable; a content generating module
for generating programmatic content in response to the
searching; and an image generation module for generating
an image as a function of the programmatic content and a
representation of the audio/video content.

15

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other aspects, features and
advantages of the present invention will be more apparent
from the following more particular description thereof,
20 presented in conjunction with the following drawings
wherein:

FIG. 1 is a block diagram showing basic
components that comprise a system for combining
video/audio content and programmatic content for storage
25 on or transmission through a medium;

FIG. 2 is a detailed block diagram illustrating
the basic components shown in FIG. 1 for combining
video/audio content and programmatic content for storage
on or transmission through a medium, such as a digital
30 video disk (DVD); and

FIG. 3 is a flow diagram depicting steps traversed in operation of the system of FIG. 1.

Corresponding reference characters indicate corresponding components throughout the several views of the drawings.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description of the presently contemplated best mode of practicing the invention is not to be taken in a limiting sense, but is made merely for the purpose of describing the general principles of the invention. The scope of the invention should be determined with reference to the claims.

Referring first to FIG. 1, a block diagram is shown of basic components that comprise a system for combining video/audio content and programmatic content for storage on or transmission through a medium. Shown is a Video/Audio authoring environment 100, authoring output 102, a programmatic content development environment 104, a software engine 106, an image 108, a formatter 110, and an output 112.

The Video/Audio Authoring Environment 100 generates the Authoring Output 102, which is provided to the software engine 106 (DVD-ROM engine) then to the Programmatic Content Development Environment 104. The Authoring Output 102 represents, distills and defines the video/audio content on a DVD, such as by representing the title and/or chapter points within the video/audio content using extensible markup language (XML). The Authoring Output 102 further includes at least one definition for at

least one variable to be included in JavaScript and HTML source files, described further hereinbelow. The Programmatic Content Development Environment 106 is used to define and/or select one or more JavaScript and HTML source files, each containing at least one variable. The DVD-ROM Engine 106 matches variables that are defined within the Authoring Output 102 to the at least one variable in the JavaScript and HTML source file in order to generate the image 108 (DVD Image file) that is passed along to the Formatter 110, which is used to generate the output 112 (DVD Output).

The Video/Audio Authoring Environment 100 may be implemented using a combination of software and a computer, such as a personal computer. The personal computer comprises at least one central processing unit (CPU), memory, a display device, and an input device, such as a keyboard. Suitable storage medium, such as a magnetic or optical disk drive, or a memory, coupled to the computer may be used to store the Authoring Output 102 upon generation, and to transfer the Authoring Output 102 to the Programmatic Content Development Environment 104 and the DVD-ROM Engine 106. Similarly, the Programmatic Content Development Environment 104 may be implemented using a combination of software and a personal computer. The computer may be as described above, with a CPU, memory, display and input device. And, suitable storage medium (such as those described above) coupled to the computer may be used to store the JavaScript and HTML source files, and to transfer the JavaScript and HTML source files to the DVD-ROM Engine. The DVD-ROM Engine,

use in embodiments and variations of the subject matter described herein without difficulty.) Shown is a DVD-Video Authoring Environment 100, XML Authoring Output 202, a DVD-ROM Engine 204, a Development Environment 206,
5 Development Utilities and Templates 208, a ROM Image 210, a DVD Formatter 212, and an Output 214.

The DVD-Video Authoring Environment 200, referred to above generically as the Video/Audio Authoring Environment, is comprised of Authoring Templates, and a
10 DVD Authoring Tool. The DVD-Video Authoring Environment 200 is supplied by third parties, such as Daikin (Scenarist and ReelDVD), Sonic Solutions (Creator and DVD-IT) and Spruce Technology (Maestro and Convergence). The DVD-Video Authoring Environment 200 provides an ability to
15 create DVD-Video and/or DVD-Audio elements (and, in some embodiments, CD-Audio elements) and provides or is modified to provide an ability to create a description of the DVD-Video and/or DVD-Audio (and/or CD-Audio) formatted using extensible markup language (XML), e.g., an XML
20 source file, i.e., the Authoring Output.

When a DVD created using the present embodiment is launched, software within the device launching the DVD checks to see if the DVD contains programmatic content. For example, if the device launching the DVD is a personal
25 computer using the Windows operating system from Microsoft, an "autorun.inf" present on the DVD may be used to launch software on the DVD that is responsive to the programmatic content on the DVD. If the DVD contains programmatic content, then the software within the device
30 launching the DVD changes to a Full Screen Channel

If a DVD author chooses, web content can be automatically displayed in the associated web window, i.e., the web content can "launch" under the control of the movie, i.e., under the control of, e.g., the DVD-, video, DVD-audio, or CD-audio content. Another option is to allow the user (at runtime) to click a button that brings up a context sensitive website. This latter option is referred to herein as "association", as opposed to, "launch".

A specific URL can be "launched" at various points within DVD-video, DVD-audio, or CD-audio content. Launching a URL has the affect of automatically (under DVD-video, DVD-audio, or CD-audio control) bringing up the associated web window, i.e., a browser, such as an

imbedded browser, and displaying content from a URL specified by the DVD-video, DVD-audio or CD-Audio. This can also occur when a user is navigating the DVD-video, DVD-audio, or CD-audio menu structures, and selects a menu item that launches a URL.

The DVD-Video Authoring Environment 200 provides an ability to add URLs to chapter points and time markers that reference the start of a PGC (or program chain). (A program chain is a section of a DVD-video title set, i.e., of a single DVD. In other words, program chains are smaller segments of video within a DVD-video title set, i.e., on a single DVD. Thus, there can be a sequence program chains on a single DVD. For "Sequential PGCs" a chapter number and PGC number are the same.)

In accordance with the present embodiment, if a button is to launch a URL, the button must be authored so that the button navigates to a chapter point. This chapter point specifies the URL to launch. In most cases, the author will choose to display a still image when the device on which the DVD is launched is not able to display DVD-ROM content in accordance with the present embodiment that indicates that web links are only functional in "DVD-ROM-equipped" DVD Players.

The concept of an "association" means that while a movie is playing, a web URL can be associated with a button at any given point. This association has a hierarchy, so that the author can for example, define an overriding URL to the DVD, and then have other URLs take precedence, as the DVD is being played or navigated. For example, the author can add default title and default

chapter URLs to the DVD URL above. At any given time, the user can click on an available AGo Online@ button, and the currently assigned URL, in accordance with the hierarchy, will be launched. Whether or not to launch a particular URL, i.e., display content at the URL in the associated web window, is under control of the user.

The Authoring Output 202 comprises an XML source file generated by the DVD-Video Authoring Environment 200. The DVD-Video Authoring Environment 200, described above, creates the XML source file 202 according to a ROM specification. This XML source file 202 gives an overview of the structure of the DVD, and indicates all events that have been specified by the author, i.e., URLs that are to be "launched" or "associated".

The DVD-ROM Engine 204 comprises runtime software (runtime binaries), an XML Parser, HTML/JAVA, and a DVD-ROM Image Engine. The DVD-ROM Engine 204 is preferably implemented as a dynamic link library (DLL) that can be called by the DVD-Video Authoring Environment. Alternatively, however, the DVD-ROM Engine 204 may be implemented as an executable, which offers that advantage that the DVD-Video Authoring Environment may be on a different computer than the DVD-ROM Engine, e.g., the DVD-Video Authoring Environment 200 may be on a Macintosh computer from Apple, while the DVD-ROM Engine may be on a "PC" operating the Windows operating system from Microsoft.

Alternatively, the DVD-ROM Engine 204 may be implemented as a browser plug-in, ActiveX control, Java class, or even as JavaScript and HTML code. Hence, as will

be appreciated by the skilled artisan, an alternative to implementing the DVD-ROM Engine as a "build time" application, as described herein, the DVD-ROM Engine 204 can be implemented as a "run time" application, wherein
5 only the XML source file and a "run time" implementation of the DVD-ROM Engine 204 are needed in order to fully realize the DVD-ROM functionality described herein, e.g., in order to "launch" and "associate" URLs at desired points within DVD title set. The functions of the DVD-ROM
10 Engine 204 implemented as a "run time" application are similar to the functions of the DVD-ROM Engine 204 implemented as a "build time" application, as described herein below, the notable exception being that the "image" generated by the "build time" application is not stored or
15 written, such as to a magnetic tape or DVD, but rather is acted upon as it is generated.

As a "build time" application, The DVD-ROM Engine 204 parses Authoring Output (from the DVD-Video Authoring Environment) and generates a template-based DVD-
20 ROM image 210 (including runtime software and programmatic content, e.g., web pages (HTML files), to be installed on the DVD), i.e., the image. The DVD-ROM image 210 preferably includes the software and programmatic content arranged in a specified directory structure that permits
25 software and programmatic content specific to particular hardware/software platforms to be located and launched upon the launching of a DVD in a particular hardware/software platform. For example, the specified directory structure (including exemplary runtime software
30 and programmatic content) may be as follows:

Additional directories, runtime software, and programmatic content are added to the above directory structure, as needed, in order to support additional hardware/software platforms, such as different types of personal computers and/or different operating systems, and consumer electronic devices, e.g., set top boxes and the like.

Thus, the DVD-ROM Engine 204 of the present embodiment offers a turnkey, simple Internet solution for all DVDs authored using the DVD-Video Authoring Environment to generate the Authoring Output that includes DVD-ROM content. (Note that as shown the Authoring Output is preferably an extensible markup language (XML) file, i.e., the XML source file, however, as will be appreciated by the person of ordinary skill in the art, and as contemplated by the inventors, the Authoring Output may be any of a number of other formats, including proprietary formats.)

The DVD-ROM Engine 204 parses the XML source file from the DVD-Video Authoring Environment 200 and generates the appropriate HTML and JavaScript code to implement Internet links, as described in further detail below. The DVD-ROM Engine 204 also combines the HTML and JavaScript with client binaries.

In practice (when the DVD-ROM Engine 204 is implemented in an executable for, as opposed to a dynamic link library (DLL)), the XML parser is activated using a command line; copies a new image of the DVD-ROM content from a golden image directory to a destination directory; extracts DVD-ROM Engine variables from the XML source file

202; and performs a search and replace of the instances of those variables in the JavaScript and HTML files in the destination directory.

Command line syntax for the XML parser, in accordance with one embodiment, is as follows:

```
itidvdrom.exe xml_source {/D destination} {/I  
golden_image} [/G graphic] [/T tag]
```

where

10

"xml_source" is a required parameter that specifies a full path of an XML source file, i.e., the Authored Output

"destination" is an optional parameter that specifies a full path to where a golden image is to be copied. If not supplied, an ITI_VAR_DESTINATION variable must be provided in the XML source file.

"golden_image" is an optional parameter that specifies a full path to the golden image of the DVD-ROM content. If not supplied, an ITI_VAR_GOLDEN_IMAGE_PATH variable must be provided in the XML source file.

"graphic" is an optional parameter that specifies the path to a graphic to be used in a DVD-ROM content installation screen. If this argument is not supplied, it can be defined in the XML source file using an ITI_VAR_STARTUP_GRAPHIC variable. If it not defined in either place, a Default DVD-ROM Engine Graphic will be used.

Engine 204 to control the copying of the software from the golden image directory (by specifying the runtime software to copy), and the substitution of variables in and the copying of the JavaScript and HTML source files.

5 The JavaScript and HTML source files contain multiple variables that are replaced with valid values prior execution of the JavaScript and HTML source files. (These JavaScript and HTML source files may comprise templates that are provided to the author in order to

10 perform functions such as providing a button associated with a URL, or, as described below may be custom JavaScript and HTML source files that are authored within the Development Environment 206.) The values of these variables are provided in the XML source file. The DVD-ROM

15 Engine 204 parses each JavaScript and HTML source file to find instances of the DVD-ROM Engine variables. When a DVD-ROM Engine variable is found, the DVD-ROM Engine looks the DVD-ROM Engine variable up in the list of variables extracted from the XML source file by the XML Parser. The

20 DVD-ROM Engine 204 then replaces the DVD-ROM Engine variable with the value and writes the file back to disk. This process is repeated for all variables in all JAVA Script and HTML source files in the DVD-ROM Engine directory.

25 The following is exemplary JavaScript code that may constitute a portion of the JavaScript and HTML source files (note that the JavaScript contains variables that will be replaced by values in the XML source file by the DVD-ROM Engine 204):

30

```

class Bookshelf {
    protected book first,last;
    protected long count;

5      Bookshelf()
    {
        first = last = null;
        count = ITI_VAR_bookcount;
    }

10     void add(book aBook)
    {
        if (first != null)
        {
15             last.next = new book(aBook);
            last = last.next;
        }
        else
        {
20             first = new book(aBook);
            last = first;
        }
        count++;
    }

25     long size() { return count; }

    void print()
    {
30         book curBook = first;
        int cnt=1;

        while (curBook != null)
        {
35             System.out.println("Book:
ITI_VAR_Temp" + cnt++);

            System.out.println("ITI_VAR_var2");
            curBook.showBook();
40             curBook = curBook.next;
        }
    }
}

45 Example

```

Using the XML source file 202 from above and the JavaScript code from above, the following JavaScript code results. Note that the variable ITI_VAR_xyz was not used in the code.

```
5  class Bookshelf {
    protected book first,last;
    protected long count;

10  Bookshelf()
    {
        first = last = null;
        count = 123;
    }

15  void add(book aBook)
    {
        if (first != null)
        {
20            last.next = new book(aBook);
            last = last.next;
        }
        else
        {
25            first = new book(aBook);
            last = first;
        }
        count++;
    }

30  long size() { return count; }

    void print()
    {
35        book curBook = first;
        int cnt=1;

        while (curBook != null)
        {
40            System.out.println("Book: abc" +
cnt++);
            System.out.println("This is a Test
");
```

```
curBook.showBook();  
curBook = curBook.next;  
    }  
5 }  
}
```

The Development Environment provides plug-ins for tools such as Visual InterDev, Dreamweaver, Homesite, and Director for HTML content development. The plug-ins assist the author in creating DVD-ROM content by automating the JavaScript creation. For example, the author may select from a menu, in one of the above mentioned tools, "insert play button", and is prompted for the graphics to use with the play button and provided with a dialog to select whether the play button is a play title, play chapter or play time, with edit boxes for the values. The Development Environment 206 also provides a user interface and emulation capabilities for building custom DVD-ROM applications, i.e., custom JavaScript and HTML source files, as mentioned above.

The Development Environment 205 preferably supports multiple languages. In order to support multiple languages, all text messages are in a configuration file. Using a "get locale" from within JavaScript code, these text messages are accessed in accordance with a desired language, so as to show the proper language. For C++ code, resource DLLs and text DLLs, per language, are employed, so that a text DLL appropriate to a desired language can be accessed in accordance with the desired language based on the Operating system language.

The Development Utilities and Templates 208 offer features for simplifying the DVD-ROM creation

process. This component includes HTML editor plug-ins,
templates for JavaScript and HTML source files,
documentation, and online developer support. The HTML
editors (i.e., the tools identified above) allow authors
5 to create the JavaScript content that will be used as
templates by the DVD-ROM Engine 204. The HTML editor
plug-ins may allow, for example, custom graphics created
by the author and code to perform DVD calls and URL
associations. By way of example, the Development Utilities
10 and Templates preferably include code for features such as
a Web button (active for, e.g., entire movie), which
allows a user to click the web button in a controller or
web page and thereby launch a desired web site; an
internet button authored into a DVD menu that launches a
15 desired web site; and Title/Chapter naming (and automatic
association with actual title/chapter numbers).

The DVD Formatter 212 calls the DVD-ROM Engine
204 and calls another engine that produces the video
assets themselves, according to the DVD being created. The
20 DVD Formatter 212 is a component of the DVD-Video
Authoring Environment but is shown as separated since the
DVD-Video and DVD-ROM images must be combined before
creating/formatting the final image/DLT (digital linear
tape).

25 FIG. 3 is a flow diagram depicting steps
traversed in operation of the system of FIG. 1.

At the outset, Video/Audio Content is authored
(Block 300) and Authoring Output is generated (Block 302)
as a function thereof. The Authoring Output, as mentioned
30 above, defines a plurality of variables that define the

Video/Audio content to be contained on the DVD. Next, Programmatic Content (DVD-ROM content) is authored (Blocks 304, 306), or appropriate templates are selected to effect DVD-ROM content for the DVD. Specifically, appropriate JavaScript and HTML source files are selected or authored, each containing at least one variable to be defined as a function of the Video/Audio content and DVD-ROM content to be placed on to the DVD. As a result of the authoring of the DVD-ROM content, software, and programmatic content are provided, and then bound with the Authoring Output (Block 308) (through the DVD-ROM Engine) in order to generate a DVD image (Block 310). The DVD image is communicated to the formatter, which formats the DVD (Block 312), creating final DVD output (Block 314).

15 While the invention herein disclosed has been
described by means of specific embodiments and
applications thereof, numerous modifications and
variations could be made thereto by those skilled in the
art without departing from the scope of the invention set
20 forth in the claims.